

МIНIСТЕРСТВО ОСВIТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота № 3**

з дисципліни “ Математичні та алгоритмічні основи компютерної графіки”

тема “ Структура файлів формату *.bmp*. Анімація примітивів за допомогою засобів бібліотеки *JavaFX* ”

|  |  |  |
| --- | --- | --- |
| Виконала  студентка III курсу  групи КП-52  Левошко Катерина Василівна  варіант № 7  Сумарний бал: |  | Перевірила  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Шкурат Оксана Сергіївна |

|  |  |
| --- | --- |
|  |  |

Київ 2018

**Мета:**

1) вивчення структури та особливостей використання файлів формату *.bmp*;

2) вивчення стандартних засобів *JavaFX* для візуалізації зображення;

3) вивчення засобів анімації примітивів в *JavaFX*.

**Завдання**

За допомогою примітивів *JavaFX* максимально реально зобразити персонажа за варіантом та виконати його *2D* анімацію. Для анімації скористатися стандартними засобами бібліотеки *JavaFX.*

Обов’язковою є реалізація таких видів анімації:

1) переміщення;

2) поворот;

3) масштабування.

Студентам пропонується скористатися розглянутими класами для читання, обробки та збереження зображень формату *.bmp* з метою використання рисунку для створення траєкторії руху або меж, в яких дозволений рух об’єктів. В даному випадку рекомендується використовувати кольори великої контрастності для різних призначень (наприклад, чорний колір відповідатиме за траєкторію руху, а інші кольори – заборонятимуть рух).

**Варіант 7:**



**Текст коду програм**

|  |
| --- |
| Heart.java |
| @Override  public void start(Stage primaryStage) {  Shadow shadow = new Shadow();  //Setting color of the shadow  shadow.setColor(Color.rgb(120, 60, 8));  //Setting the height of the shadow  shadow.setHeight(5);  //Setting the width of the shadow  shadow.setWidth(5);  //Setting the radius of the shadow  shadow.setRadius(5);  Group root = new Group();  Scene scene = new Scene(root, 1200, 600);  // Создаём фигуру heart  Path heart = new Path();  // Создаём левую и правую части heart  MoveTo moveToLeft = new MoveTo(245.0, 180.0);  CubicCurveTo leftCurve = new CubicCurveTo  (111.0f, 110.0f, 74.0f, 231.0f, 233.0f, 423.0f);  MoveTo moveToRight = new MoveTo(245.0, 180.0);  CubicCurveTo rightCurve = new CubicCurveTo  (404.0f, 100.0f, 374.0f, 297.0f, 233.0f, 423.0f );  heart.getElements().addAll(moveToLeft, leftCurve, moveToRight, rightCurve);  heart.setFill(Color.rgb(255, 0, 0));  root.getChildren().add(heart);  Path around = new Path();  MoveTo upMoveTo = new MoveTo(27.0, 214.0);  CubicCurveTo upCurve = new CubicCurveTo  (121.0f, 289.0f, 365.0f, 140.0f, 458.0f, 214.0f);  VLineTo leftLine = new VLineTo(289.0);  MoveTo downMoveTo = new MoveTo(458.0, 289.0);  CubicCurveTo downCurve = new CubicCurveTo  (365.0f, 215.0f, 121.0f, 364.0f, 27.0f, 289.0f);  VLineTo rightLine = new VLineTo(214.0);  around.getElements().addAll(upMoveTo,upCurve, leftLine, downMoveTo, downCurve, rightLine);  around.setFill(Color.rgb(181, 115, 54));  root.getChildren().add(around);  QuadCurve rightQuad = new QuadCurve  (27.0, 214.0, 49.0, 198.0, 130.0, 200.0);  QuadCurve leftQuad = new QuadCurve  (458.0, 289.0, 406.0, 316.0, 332.0, 290.0);  rightQuad.setFill(Color.rgb(181, 115, 54));  rightQuad.setEffect(shadow);  leftQuad.setFill(Color.rgb(181, 115, 54));  leftQuad.setEffect(shadow);  root.getChildren().addAll(rightQuad, leftQuad);  Line line1 = new Line(370, 110, 287, 194);  line1.setStroke(Color.rgb(176, 122, 28));  Line line2 = new Line(175, 316, 120, 376);  line2.setStroke(Color.rgb(176, 122, 28));  line1.setStrokeWidth(4.0f);  line2.setStrokeWidth(4.0f);  Polygon bigTriangle = new Polygon(100, 360, 135, 395, 85, 425);  bigTriangle.setFill(Color.rgb(194, 133, 55));  Circle circle = new Circle(120.0, 376.0, 7.0);  circle.setFill(Color.rgb(213, 162, 60));  Polygon uppol = new Polygon(373, 110, 327, 153, 323, 101, 370, 45);  Polygon downpol = new Polygon(373, 110, 327, 153, 385, 165, 431, 126);  uppol.setFill(Color.rgb(213, 162, 60));  downpol.setFill(Color.rgb(213, 162, 60));  root.getChildren().addAll(line1, line2, circle, bigTriangle, uppol, downpol);  int cycleCount = 2; //  int time = 4000;  ScaleTransition scaleTransition = new ScaleTransition(Duration.millis(time), root);  scaleTransition.setToX(2);  scaleTransition.setToY(2);  scaleTransition.setCycleCount(cycleCount);  scaleTransition.setAutoReverse(true);  TranslateTransition translateTransition = new TranslateTransition(Duration.millis(time), root);  translateTransition.setFromX(50);  translateTransition.setToX(750);  translateTransition.setCycleCount(cycleCount+2);  translateTransition.setAutoReverse(true);  RotateTransition rotateTransition = new RotateTransition(Duration.millis(time), root);  rotateTransition.setByAngle(180f);  rotateTransition.setCycleCount(cycleCount);  rotateTransition.setAutoReverse(true);  ScaleTransition scaleTransition2 = new ScaleTransition(Duration.millis(time), root);  scaleTransition2.setToX(0.5);  scaleTransition2.setToY(0.5);  scaleTransition2.setCycleCount(cycleCount);  scaleTransition2.setAutoReverse(true);  ParallelTransition parallelTransition = new ParallelTransition();  parallelTransition.getChildren().addAll(  translateTransition,  scaleTransition,  rotateTransition,  scaleTransition2  );  parallelTransition.setCycleCount(Timeline.INDEFINITE);  parallelTransition.play();  primaryStage.setResizable(false);  primaryStage.setTitle("Heart");  primaryStage.setScene(scene);  primaryStage.show();  } |

**Висновки**

Виконавши дану лабораторну роботу я вивчила структуру та особливості використання файлів формату *.bmp*, вивчила стандартні засоби *JavaFX* для візуалізації зображення, вивчила засоби анімації примітивів в *JavaFX*.